Université de Montréal
C.P. 6128, Succursale Centre-ville,
Montréal (Québec) Canada H3C 3J7

Téléphone : +1-514-343-7575
Télécopieur (Fax) : +1-514-343-7121
crt@crt.umontreal.ca
www.crt.umontreal.ca/CRT/

CRT-2000-04

Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

A TABU SEARCH ALGORITHM FOR THE SITE DEPENDENT VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

by

Jean-François Cordeau
Gilbert Laporte

January 2000

101-01785

PUBLICATION
CRT-2000-04

*A TABU SEARCH ALGORITHM FOR THE SITE DEPENDENT VEHICLE ROUTING PROBLEM WITH TIME WINDOWS**

by

Jean-François Cordeau[1]
Gilbert Laporte[1,2]

* *This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). This support is gratefully acknowledged.*

[1] École des Hautes Études Commerciales de Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, CANADA H3T 2A7

[2] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, CANADA H3C 3J7

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2000

Abstract

This paper shows that the Site Dependent Vehicle Routing Problem (SDVRP) can be viewed as a special case of the Periodic VRP. A tabu search heuristic for the SDVRP and the SDVRP with Time Windows is outlined and tested on randomly generated instances. In the case of the SDVRP for which benchmark results are available, the proposed method outperforms all previously published algorithms.

Keywords: site dependent vehicle routing problem; time windows; tabu search.

Résumé

Cet article montre que le Problème de Tournées de Véhicules avec Contraintes de Site (PTVCS) peut être vu comme un cas particulier du Problème de Tournées de Véhicules Périodique. Une heuristique de recherche avec tabous pour le PTVCS et le PTVCS avec fenêtres de temps est décrite sommairement et testée sur des instances générées aléatoirement. Dans le cas du PTVCS pour lequel des résultats comparatifs sont disponibles, la méthode obtient de meilleurs résultats que ceux produits par toutes les méthodes existantes.

Mots-clés: problème de tournées de véhicules avec contraintes de site; fenêtres de temps; recherche avec tabous.

1 Introduction

The *Site Dependent Vehicle Routing Problem* (SDVRP) is defined on a graph $G = (V, A)$, where $V = \{v_0, v_1, \ldots, v_n\}$ is the vertex set and $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ is the arc set. Vertex v_0 is a depot at which is based a fleet composed of m vehicles of t different types. For each vehicle type ℓ ($\ell = 1, \ldots, t$), m_ℓ vehicles are available and each has a capacity Q_ℓ. Every vertex of $V \setminus \{v_0\}$ corresponds to a city or a customer and has a nonnegative demand q_i, as well as a nonnegative service time d_i. In addition, every customer v_i can only be served by a subset of vehicle types $R_i \subseteq \{1, \ldots, t\}$, with $|R_i| = f_i$. With A is associated a cost or travel time matrix $C = (c_{ij})$. The SDVRP consists of designing a set of m vehicle routes on G such that (i) each route starts and ends at the depot; (ii) each customer belongs to exactly one route; (iii) each customer v_i is served by a vehicle type belonging to R_i; (iv) the total demand of the route followed by a vehicle of type ℓ does not exceed Q_ℓ, and its total duration does not exceed a preset value D_ℓ; and (v) the total duration of all routes is minimized. An extension of the SDVRP, called the *SDVRP with Time Windows* (SDVRPTW) is obtained by imposing that the service time of customer i should start within the time window $[e_i, l_i]$. As is standard in such problems, a vehicle may arrive before e_i at the location of customer v_i and only start service at time e_i. The SDVRP is NP-hard and includes as a special case the classical *Vehicle Routing Problem* (VRP) if there is a single type of vehicle and the *Heterogeneous Fleet VRP* if $R_i = \{1, \ldots, t\}$ for every customer v_i.

The SDVRP can also be viewed as a special case of the *Periodic VRP* (PVRP), another VRP extension. This problem works with a planning horizon of h days and each customer v_i specifies the number a_i of visits it requires, as well as a set S_i of allowable combinations. For example, if $h = 5$, $a_i = 3$ and $S_i = \{\{1, 2, 4\}, \{2, 3, 5\}\}$, then customer v_i requires three visits, and these must take place on days 1, 2 and 4, or on days 2, 3 and 5. The PVRP consists of simultaneously selecting an allowable combination of visit days for each customer, as well as a set of vehicle routes for each day according to the VRP rules. The SDVRP is transformed into a PVRP by simply setting $h = t$, and $a_i = 1$ and $S_i = \{\{\ell\} : \ell \in R_i\}$ for every customer v_i. In this article, we exploit this relationship. In particular, we show that a tabu search heuristic previously developed by CORDEAU et al. (1997) for the PVRP can be

applied to the SDVRP without a single modification, and the results obtained outperform those of all previously published heuristics for the SDVRP. In addition, an algorithm for the SDVRPTW is obtained by using the adaptation of this heuristic that was later introduced by CORDEAU et al. (2000) to handle time windows. We first present in Section 2 an overview of the heuristic, followed by computational experiments in Section 3.

2 Overview of the Heuristic

The PVRP heuristic whose detailed description is provided in CORDEAU et al. (1997) is based on tabu search, a metaheuristic first proposed by GLOVER (1986). We now summarize it in the context of the SDVRP.

An initial solution is obtained by first randomly assigning to each customer v_i a vehicle type in R_i, and then constructing a VRP solution for each vehicle type taken separately. This is done by gradually inserting customers into partially constructed vehicle routes by means of the generalized insertion heuristic (GENI) initially proposed by GENDREAU et al. (1992). This process does not ensure that all vehicle routes are feasible with respect to capacity and duration constraints.

In the improvement phase, an attempt is made to minimize a function equal to the routing cost, plus two weighted penalty terms representing the overcapacity and overduration of an infeasible solution. These factors are dynamically adjusted throughout the search. At every iteration, the neighbourhood of a solution is obtained by performing one of the following exchanges : (a) remove a customer v_i from its current route and insert it into the route of another vehicle of the same type; (b) remove a customer v_i from its current route and insert it into the route of a vehicle of a different type in R_i. In an exchange of type (b), the new vehicle is chosen so as to minimize the insertion cost. Whenever a customer is removed from a route, reinserting it into that route is forbidden for the next θ iterations, where θ is a user controlled parameter. To diversify the search, the cost of a solution leading to an increase in the objective function is biased so as to penalize frequently performed moves. The search stops after a set number of iterations.

In the SDVRPTW, time windows are handled by adding to the objective function an extra penalty term representing time window violations. The remainder of the algorithm, which is described in detail in CORDEAU et al. (2000) is however very similar.

3 Computational Experiments

To test our approach, we first generated one set of new instances for the SDVRP and two sets for the SDVRPTW. Instances for the SDVRP were generated randomly by using a procedure previously described by CORDEAU et al. (1997), that creates clusters of customers around a certain number of seed points. A total of ten instances were generated, ranging in size from 48 to 288 customers and having four or six types of vehicles. For the instances with $t = 4$, the number f_i of different vehicle types that can visit customer v_i is either 1, 2 or 4. If $f_i = 1$, then the single element of set R_i is chosen randomly in $\{1, 2, 3, 4\}$. If $f_i = 2$, then the set R_i can be either $\{1, 2\}$, $\{2, 3\}$, $\{3, 4\}$ or $\{1, 4\}$ and each possibility has an equal probability of being chosen. For the instances with $t = 6$, the value of f_i is either 1, 3 or 6, and if $f_i = 3$, then R_i can be either $\{1, 2, 3\}$, $\{2, 3, 4\}$, $\{3, 4, 5\}$, $\{4, 5, 6\}$, $\{1, 5, 6\}$ or $\{1, 2, 6\}$.

Two groups of instances for the SDVRPTW were then created by adding time windows to the newly generated SDVRP instances. In group (a), narrow time windows were added by first choosing a uniform random number e_i in the interval $[60, 480]$ and then choosing a uniform random number l_i in the interval $[e_i + 90, e_i + 180]$. In group (b), larger time windows were created by choosing the random numbers e_i and l_i in the intervals $[60, 300]$ and $[e_i + 180, e_i + 360]$, respectively. In all instances, the depot has a time window of $[0, 1000]$.

The characteristics of the newly generated test instances are summarized in Table 1. The values m and m' indicate the number of vehicles of each type that are available in the SDVRP and SDVRPTW instances, respectively. The capacity Q_ℓ of a vehicle varies from one type ℓ to another. However, all routes have the same limit D on total duration. All data are available from the Internet at `http://www.crt.umontreal.ca/~cordeau/tabu/`.

Table 1: Characteristics of new instances for the SDVRP and SDVRPTW

No.	n	t	m	m'	D	Q_ℓ
1	48	4	1	2	500	100, 150, 200, 250
2	96	4	2	3	500	100, 150, 200, 250
3	144	4	3	4	500	100, 150, 200, 250
4	192	4	4	5	500	100, 150, 200, 250
5	240	4	5	6	500	100, 150, 200, 250
6	288	4	6	7	500	100, 150, 200, 250
7	72	6	1	2	500	125, 150, 175, 200, 225, 250
8	144	6	2	3	500	125, 150, 175, 200, 225, 250
9	216	6	3	4	500	125, 150, 175, 200, 225, 250
10	288	6	4	5	500	125, 150, 175, 200, 225, 250

Tables 2 and 3 report the results obtained by executing the algorithm once on every instance using standard parameter settings (CORDEAU et al., 1997). Column $c(s^*)$ indicates the cost of the best solution found while columns T^* and T indicate the CPU time needed to identify this solution and the total CPU time needed for executing the algorithm. The two columns $c(s^{**})$ and *Gap* indicate the cost of the best solution found during all experiments and the gap between this solution and that found with standard parameter settings. The program was coded in C and experiments were performed on a Sun Ultra 2 (300 MHz).

Table 2: Results on the new SDVRP instances

No.	n	t	m	$c(s^*)$	T^*	T	$c(s^{**})$	*Gap* (%)
1	48	4	1	1385.47	0.85	1.92	1384.15	0.10
2	96	4	2	2322.08	2.54	4.27	2320.97	0.05
3	144	4	3	2626.97	2.19	7.12	2623.31	0.14
4	192	4	4	3575.79	8.85	10.10	3500.79	2.14
5	240	4	5	4479.34	12.90	13.18	4479.34	0.00
6	288	4	6	4583.46	11.16	15.84	4546.79	0.81
7	72	6	1	1990.68	0.20	3.39	1955.11	1.82
8	144	6	2	3103.40	2.88	7.31	3082.32	0.68
9	216	6	3	3732.06	6.86	13.47	3664.22	1.85
10	288	6	4	4782.17	1.09	16.03	4739.43	0.90

Table 3: Results on the new SDVRPTW instances

No.	n	t	m	$c(s^*)$	T^*	T	$c(s^{**})$	Gap (%)
1a	48	4	2	1658.84	1.88	4.44	1655.42	0.21
2a	96	4	3	3003.21	10.44	12.64	2975.18	0.94
3a	144	4	4	3450.46	20.02	23.82	3373.52	2.28
4a	192	4	5	4746.2[illegible]	33.48	36.65	4682.84	1.35
5a	240	4	6	6521.95	4.14	43.07	6278.89	3.73
6a	288	4	7	6066.22	51.31	63.30	6014.95	0.85
7a	72	6	2	2179.06	7.17	7.69	2179.06	0.00
8a	144	6	3	4029.71	0.73	21.72	3988.18	1.04
9a	216	6	4	5085.57	14.10	39.22	4982.19	2.07
10a	288	6	5	6218.15	24.12	60.53	6050.41	2.77
1b	48	4	2	1429.35	0.51	7.11	1429.35	0.00
2b	96	4	3	2537.07	1.39	13.53	2494.54	1.70
3b	144	4	4	2850.20	20.27	27.53	2807.69	1.51
4b	192	4	5	3828.65	17.85	40.32	3775.86	1.40
5b	240	4	6	4842.61	30.63	52.47	4721.91	2.56
6b	288	4	7	5039.36	34.48	71.90	4934.40	2.13
7b	72	6	2	1872.98	7.62	11.21	1850.44	1.22
8b	144	6	3	3190.48	23.15	29.15	3175.25	0.48
9b	216	6	4	4081.26	44.44	46.28	4017.27	1.59
10b	288	6	5	5298.60	22.19	71.65	5159.84	2.69

We then compared our algorithm with previously proposed methods for the SDVRP. Only few benchmark results exist for this problem. The best are those of CHAO et al. (1999) which themselves outperform those of NAG et al. (1988). As far as we are aware, no results are available for the SDVRPTW.

Using the same parameter settings as in our first set of experiments, we tested our heuristic on a set of 23 instances introduced by CHAO et al. (1999). Table 4 reports the results obtained on these instances. In instances 7 to 18, the difference in cost between our solutions and those of CHAO et al. (1999) is sometimes explained by rounding errors. In their article, CHAO et al. (1999) mention that the Euclidean coordinates for customer and depot locations should be rounded after the fourth decimal. However, in the data that was provided to us by the authors, these values were rounded after the first decimal. Since there are no duration constraints in these instances and the volume at each customer is an integer,

the rounding affects only the evaluation of the cost and not the feasibility of the solutions. These results show that without any parameter tuning, our algorithm always finds better solutions than those reported by CHAO et al. (1999). Since these authors performed their experiments on a computer which is roughly two times slower than the one we used, it is fair to say that our heuristic requires sligthly longer CPU times. Finally, the best solution that we report show considerable improvement over the previous best known solutions.

Table 4: Results on SDVRP instances from the literature

	Characteritics			CGW		CL			
No.	n	t	m_t	$c(s^*)$	T	$c(s^*)$	T^*	T	$c(s^{**})$
1	50	3	3,5,2	668.58	0.74	643.80	1.27	1.87	642.66
2	50	2	5,3	610.04	0.47	598.10	0.50	1.99	598.10
3	75	3	7,6,3	1002.45	2.55	977.40	3.10	3.29	959.36
4	75	2	6,5	900.93	3.67	861.93	2.87	3.12	854.43
5	100	3	4,6,5	1071.54	5.98	1025.84	3.21	4.53	1020.22
6	100	2	8,4	1080.89	3.32	1053.03	0.46	4.98	1036.02
7	27	3	1,1,1	391.30	0.03	391.18	0.01	0.97	391.18
8	54	3	2,2,2	664.46	0.08	664.48	0.02	2.15	664.48
9	81	3	3,3,3	948.23	0.46	948.00	2.78	3.21	948.00
10	108	3	4,4,4	1252.57	1.40	1238.69	3.84	4.82	1238.69
11	135	3	5,5,5	1526.60	6.76	1497.31	6.05	6.49	1469.97
12	162	3	6,6,6	1854.82	7.55	1740.43	0.54	8.10	1729.70
13	54	3	2,2,2	1205.53	0.15	1196.46	0.37	2.11	1196.46
14	108	3	4,4,4	2092.68	1.24	2057.49	2.25	4.35	2028.06
15	162	3	6,6,6	2966.77	2.57	2870.57	0.76	7.97	2754.79
16	216	3	8,8,8	3710.96	12.67	3515.50	9.13	11.52	3479.81
17	270	3	10,10,10	4441.53	9.53	4260.22	13.25	14.39	4260.22
18	324	3	12,12,12	5085.28	29.50	5017.53	18.85	19.90	4877.49
19	100	3	2,4,2	878.58	10.38	852.05	3.01	5.27	850.39
20	150	3	3,6,3	1126.80	30.40	1055.35	7.43	8.40	1046.14
21	199	3	4,9,4	1420.85	38.29	1338.65	8.66	11.56	1337.83
22	120	3	2,3,2	1150.13	2.23	1060.45	3.11	6.39	1012.87
23	100	3	3,4,3	837.98	1.99	819.92	1.20	4.70	818.75

Acknowledgements

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada under grant OGP0039682. This support is gratefully acknowledged. Thanks are also due to Edward A. Wasil who provided the CGW test data.

References

I-M. Chao, B. L. Golden and E. A. Wasil. "A Computational Study of a New Heuristic for the Site-Dependent Vehicle Routing Problem." *INFOR*, **37**:319–336 (1999).

J.-F. Cordeau, G. Laporte and M. Gendreau. "A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems." *Networks*, **30**:105–119 (1997).

J.-F. Cordeau, G. Laporte and A. Mercier. "A Unified Tabu Search Heurisitc for Vehicle Routing Problems with Time Windows." Technical Report CRT-00-03, Centre for Research on Transportation, Montréal, 2000.

M. Gendreau, A. Hertz and G. Laporte. "New Insertion and Postoptimization Procedures for the Traveling Salesman Problem." *Operations Research*, **40**:1086–1094 (1992).

F. Glover. "Future paths for integer programming and links to artificial intelligence." *Computers & Operations Research*, **13**:533–549 (1986).

B. Nag, B. Golden and A. Assad. "Vehicle Routing with Site Dependencies." In B. L. Golden and A. A. Assad, editors, *Vehicle Routing: Methods and Studies*, pages 149–159. North-Holland, Amsterdam, 1988.